

SPRAWOZDANIE

Autor: Michał Deć

1. Wstęp.....	2
2. Opis implementacji problemu i podstaw zagadnienia.....	2
3. Podział programu.....	2
3.1. Odczyt danych wejściowych.....	2
3.2. Sortowanie.....	2
3.3. Zapis wyniku.....	3
4. Schemat blokowy algorytmu.....	4
5. Znane ograniczenia programu.....	8
6. Uruchomienie programu i sprawdzenie wyników.....	8
7. Podsumowanie.....	9

Rzeszów, rok 2021

1. Wstęp

Program ma za zadanie wyszukanie największej możliwej liczby, konstruując ją poprzez złączenie podanych uprzednio liczb.

Przykład:

Wejście: [10, 81, 9]

Wyjście: 98110

2. Opis implementacji problemu i podstaw zagadnienia

Należy zauważyć, że aby stworzyć największą możliwą liczbę z uprzednio podanych liczb, to należy te liczby ustawić w takiej kolejności aby każda z kolejnych liczb sąsiadujących ze sobą również dawała największą możliwą wartość.

Przykładowo w tablicy [91, 8, 54, 10] kombinacja 9185410 daje największą możliwą liczbę, jeżeli każda para sąsiadujących ze sobą liczb daje również największą możliwą liczbę:

918 > 891

854 > 548

5410 > 1054

Jeżeli przykładowo w tablicy [7, 9, 10] kombinacja 97 daje większą wartość niż 79, to pozycje tych liczb należy zamienić w tablicy, aby utworzyć [9, 7, 10].

3. Podział programu

Program składa się z czterech funkcji, których zastosowanie zostało opisane poniżej:

- void wczytanie_danych_wejscowych(void)
- int main()
- int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)
- void zapis_wyniku(void)

3.1. Odczyt danych wejściowych

Wszystkie liczby, z których powinna zostać utworzona największa możliwa liczba powinny zostać zapisane w pliku tekstowym **dane_wejscowe.txt**. Każda z liczb powinna zostać oddzielona spacją.

Za odczyt danych wejściowych odpowiada funkcja **void wczytanie_danych_wejscowych(void)**. Najpierw funkcja ta sprawdza ile liczb jest zapisanych w pliku tekstowym **dane_wejscowe.txt**, a następnie dokonuje alokacji miejsca na tablice danych, której ilość elementów odpowiada ilości liczb z pliku tekstowego. Następnie funkcja ta odczytuje kolejno liczby z pliku tekstowego i umieszcza je w tablicy danych.

3.2. Sortowanie

Za sortowanie liczb w tablicy danych odpowiadają funkcje **int main()** oraz **int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)**.

Za sprawdzenie która kombinacja dwóch sąsiadujących ze sobą liczb daje większą wartość odpowiada funkcja **int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)**, do której przesyłane są kolejno po dwie sąsiadujące ze sobą liczby z tablicy danych. Powyższa funkcja zwraca wartość:

- 0, kiedy kombinacja <pierwsza_liczba><druga_liczba> daje większą wartość niż <druga_liczba><pierwsza_liczba>, co oznacza że druga liczba nie jest bardziej znacząca,
- 1, w odwrotnej sytuacji, kiedy kombinacja <druga_liczba><pierwsza_liczba> daje większą wartość niż <pierwsza_liczba><druga_liczba>, czyli w sytuacji kiedy druga liczba jest bardziej znacząca.

Na podstawie zwróconego wyniku funkcja **int main()** podejmuje decyzje czy pozycje dwóch sąsiadujących liczb należy zamienić. Jeśli wynikiem działania funkcji **int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)** jest wartość:

- 0, to dwie sąsiadujące ze sobą liczby należy zostawić na aktualnych pozycjach, gdyż druga z sąsiadujących liczb nie jest bardziej znacząca,
- 1, to pozycje dwóch sąsiadujących ze sobą liczb należy zamienić, gdyż druga liczba jest bardziej znacząca.

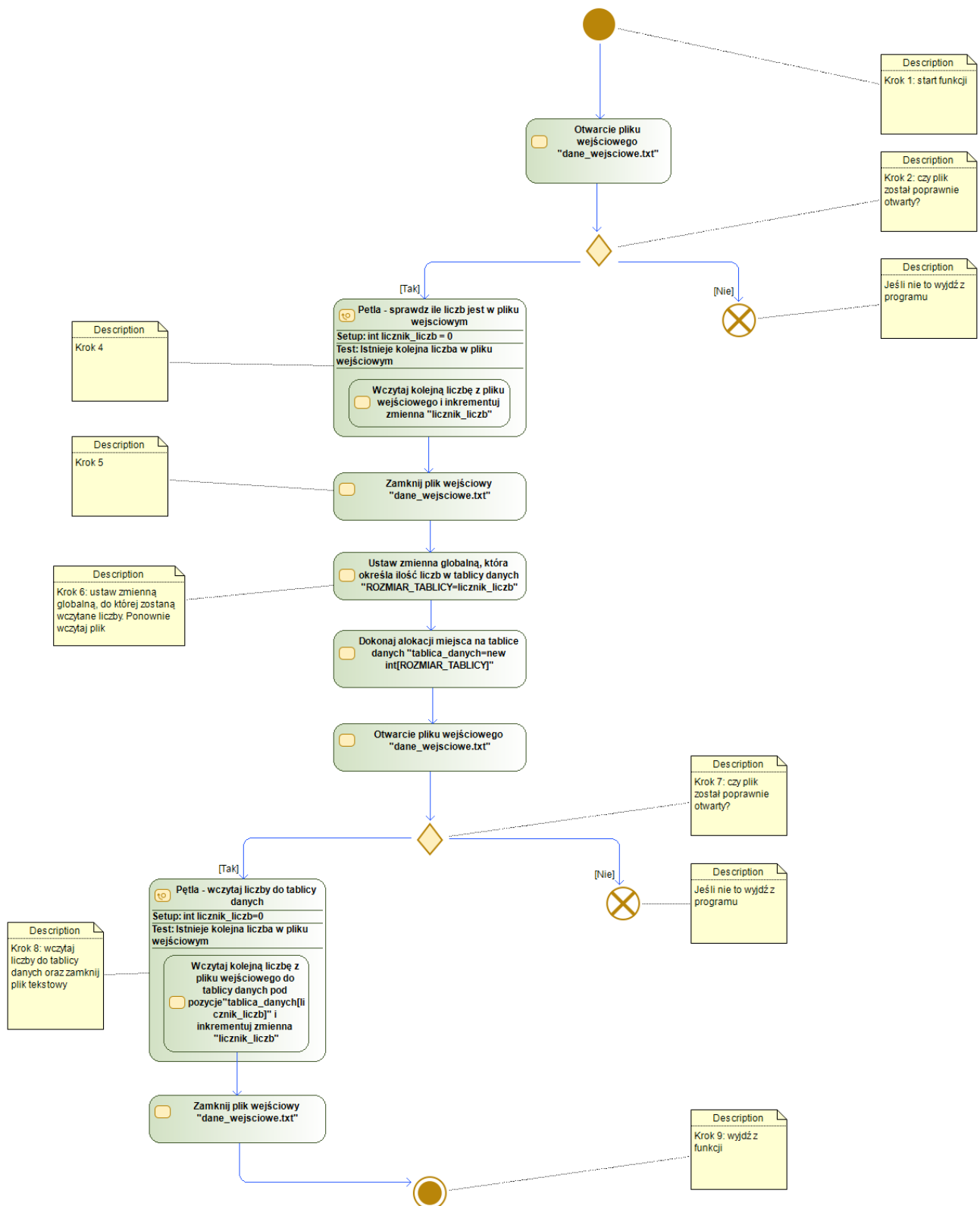
Funkcja **int main()** wywołuje w pętli funkcję **int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)**, przekazując do niej kolejno następne sąsiadujące ze sobą liczby z tablicy danych, do momentu kiedy żadna para sąsiadujących ze sobą liczb w tablicy nie będzie wymagała zamiany pozycji, co oznacza że wszystkie liczby ułożono już na pozycjach, które w wyniku dają największą możliwą liczbę.

3.3. Zapis wyniku

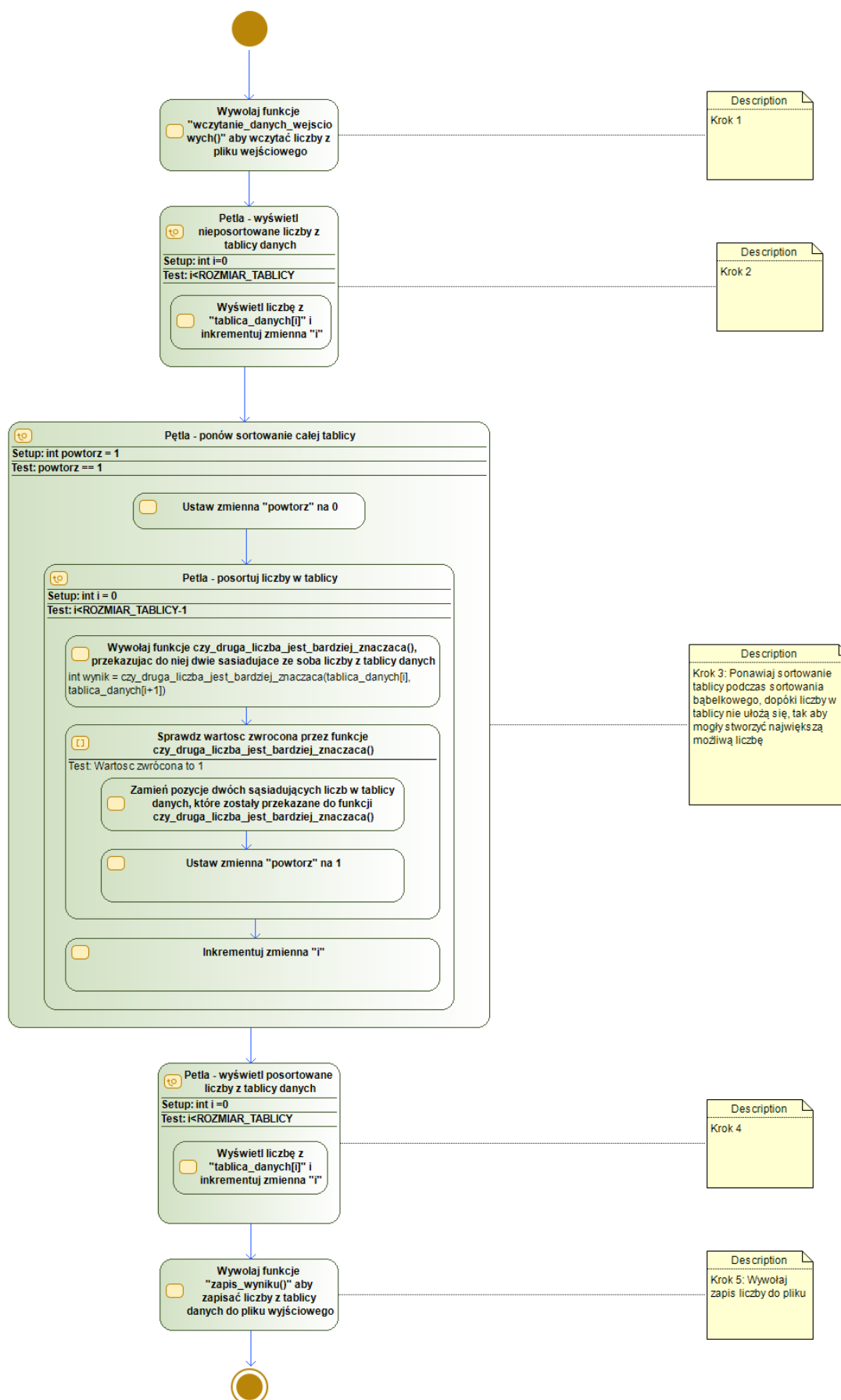
Funkcja **void zapis_wyniku(void)** jest odpowiedzialna za zapisanie liczb z posortowanej tablicy danych do pliku wyjściowego **wynik.txt**.

4. Schemat blokowy algorytmu

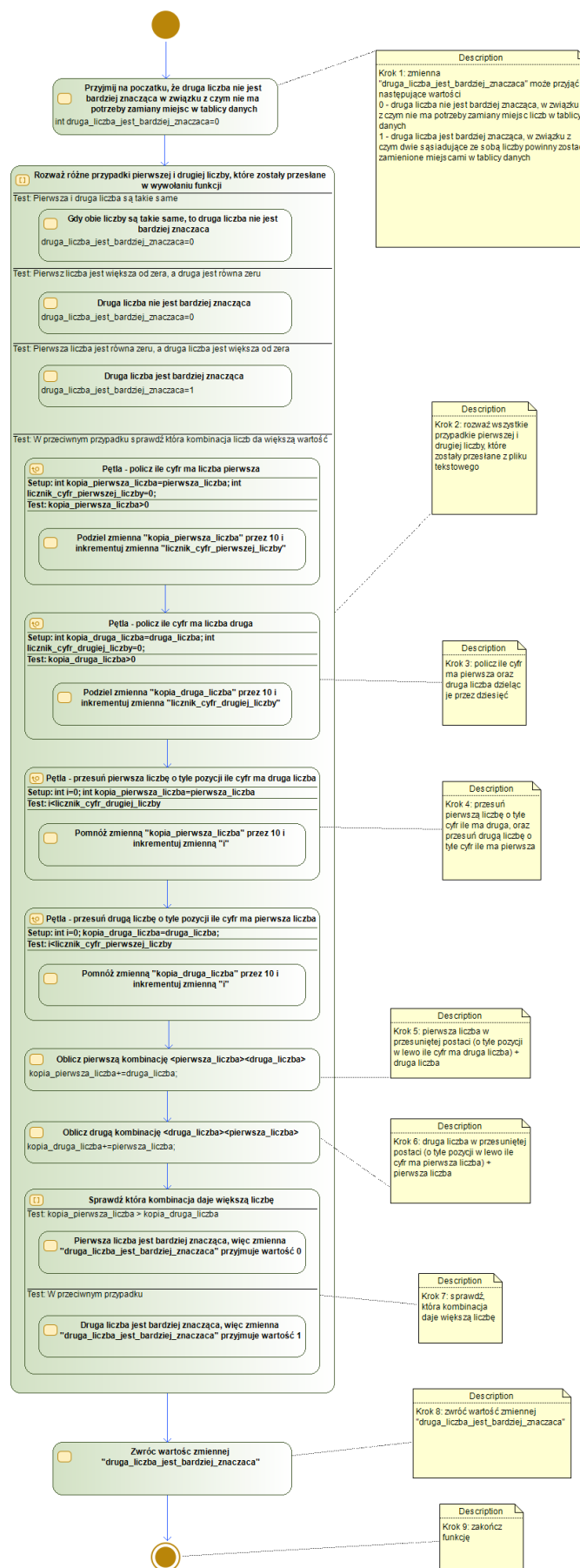
Funkcja void wczytanie_danych_wejscowych(void)



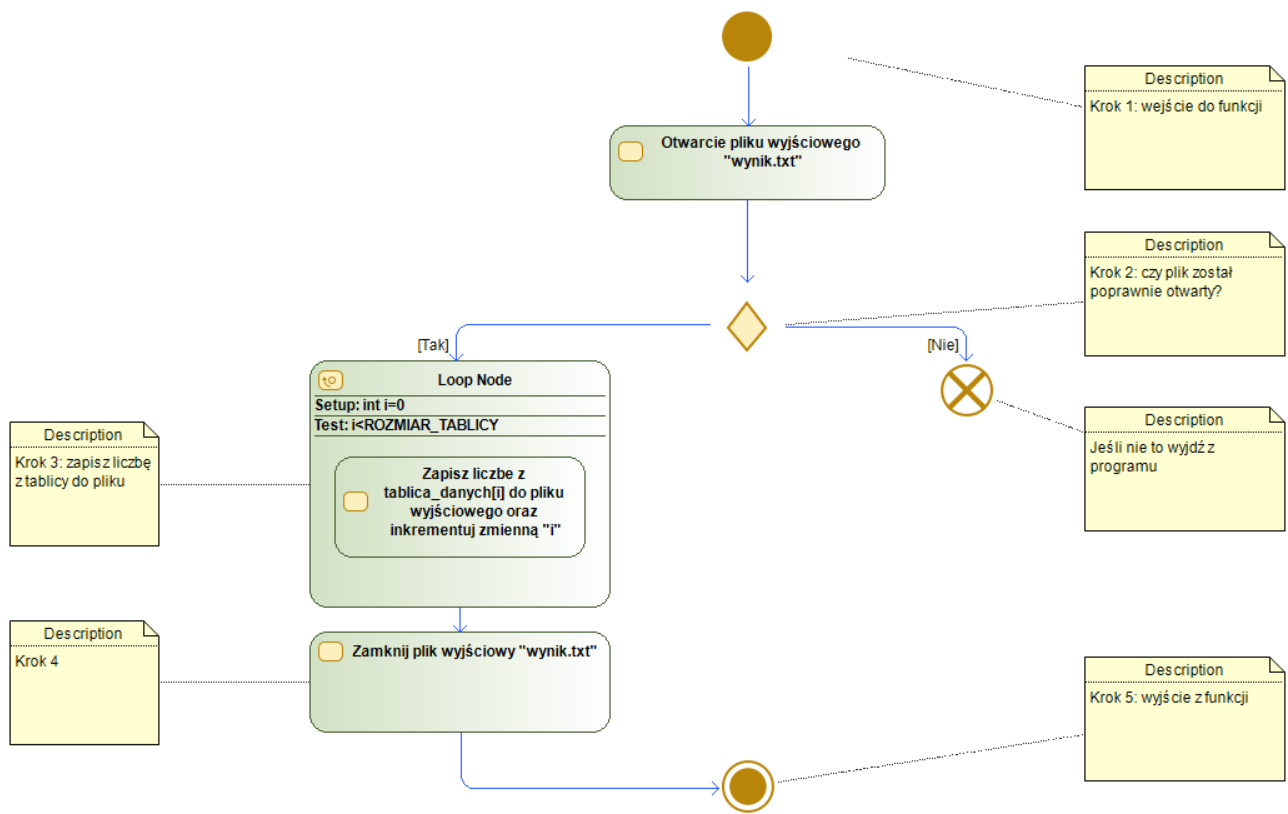
Funkcja int main()



Funkcja int czy_druga_liczba_jest_bardziej_znaczaca (int pierwsza_liczba, int druga_liczba)



Funkcja void zapis_wyniku(void)



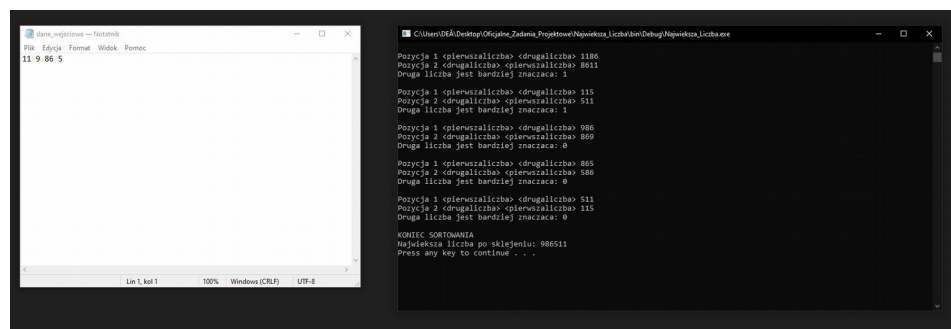
5. Znane ograniczenia programu

Program wczytuje liczby z pliku wejściowego do zmiennych typu int oraz oczekuje tylko liczb całkowitych dodatnich lub zera.

6. Uruchomienie programu i sprawdzenie wyników

Przykłady testów wykonanych wewnątrz programu:

Test 1:



The screenshot shows two windows. The left window is a Notepad++ editor with the file 'dane_wejscowe.txt' containing the text '11 9 86 5'. The right window is a debugger console showing the execution of 'Najwieksza_Liczba.exe'. The output includes comparisons between pairs of numbers (e.g., 1180 and 8611) and a final result of 986511.

```
pozycja 1 <pierwszaliczba> <drugialiczba> 1180
pozycja 2 <drugialiczba> <pierwszaliczba> 8611
Druga liczba jest bardziej znacząca: 1

pozycja 1 <pierwszaliczba> <drugialiczba> 115
pozycja 2 <drugialiczba> <pierwszaliczba> 911
Druga liczba jest bardziej znacząca: 1

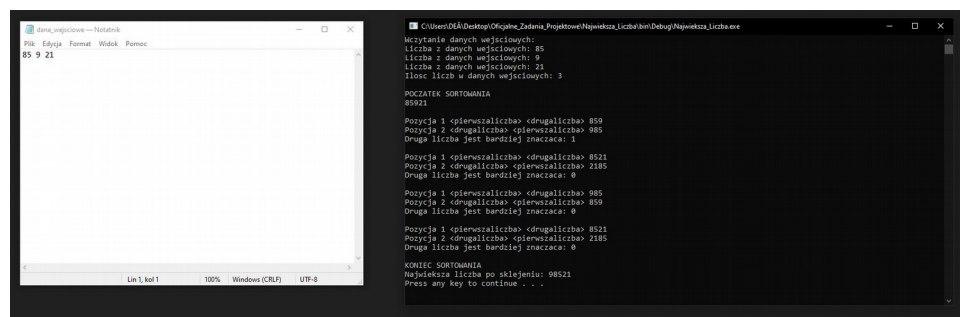
pozycja 1 <pierwszaliczba> <drugialiczba> 986
pozycja 2 <drugialiczba> <pierwszaliczba> 869
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 865
pozycja 2 <drugialiczba> <pierwszaliczba> 586
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 511
pozycja 2 <drugialiczba> <pierwszaliczba> 115
Druga liczba jest bardziej znacząca: 0

KONIEC SORTOWANIA
Najwieksza liczba po sklejeniu: 986511
Press any key to continue . . .
```

Test 2:



The screenshot shows two windows. The left window is a Notepad++ editor with the file 'dane_wejscowe.txt' containing the text '85 9 21'. The right window is a debugger console showing the execution of 'Najwieksza_Liczba.exe'. The output includes the number of input numbers (3), a sorting process, and a final result of 98521.

```
Wczytanie danych wejściowych:
Liczba 1 danych wejściowych: 85
Liczba 2 danych wejściowych: 9
Liczba 3 danych wejściowych: 21
Ilość liczb w danych wejściowych: 3

POCZĄTEK SORTOWANIA
85921

pozycja 1 <pierwszaliczba> <drugialiczba> 859
pozycja 2 <drugialiczba> <pierwszaliczba> 985
Druga liczba jest bardziej znacząca: 1

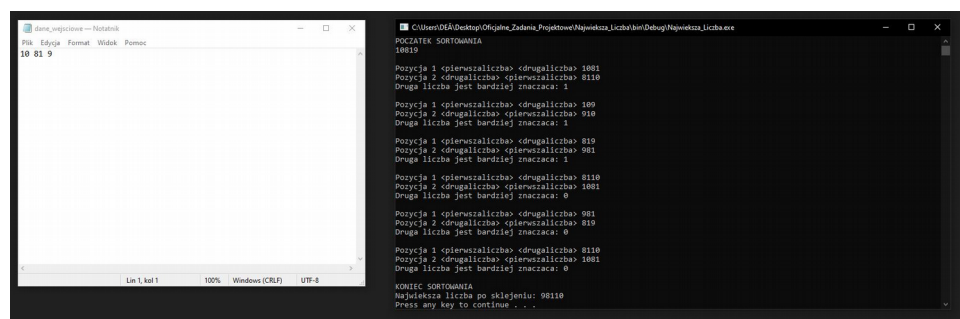
pozycja 1 <pierwszaliczba> <drugialiczba> 8521
pozycja 2 <drugialiczba> <pierwszaliczba> 2185
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 985
pozycja 2 <drugialiczba> <pierwszaliczba> 859
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 8521
pozycja 2 <drugialiczba> <pierwszaliczba> 2185
Druga liczba jest bardziej znacząca: 0

KONIEC SORTOWANIA
Najwieksza liczba po sklejeniu: 98521
Press any key to continue . . .
```

Test 3:



The screenshot shows two windows. The left window is a Notepad++ editor with the file 'dane_wejscowe.txt' containing the text '10 81 9'. The right window is a debugger console showing the execution of 'Najwieksza_Liczba.exe'. The output includes a sorting process and a final result of 98110.

```
POCZĄTEK SORTOWANIA
10819

pozycja 1 <pierwszaliczba> <drugialiczba> 1081
pozycja 2 <drugialiczba> <pierwszaliczba> 8110
Druga liczba jest bardziej znacząca: 1

pozycja 1 <pierwszaliczba> <drugialiczba> 189
pozycja 2 <drugialiczba> <pierwszaliczba> 918
Druga liczba jest bardziej znacząca: 1

pozycja 1 <pierwszaliczba> <drugialiczba> 819
pozycja 2 <drugialiczba> <pierwszaliczba> 981
Druga liczba jest bardziej znacząca: 1

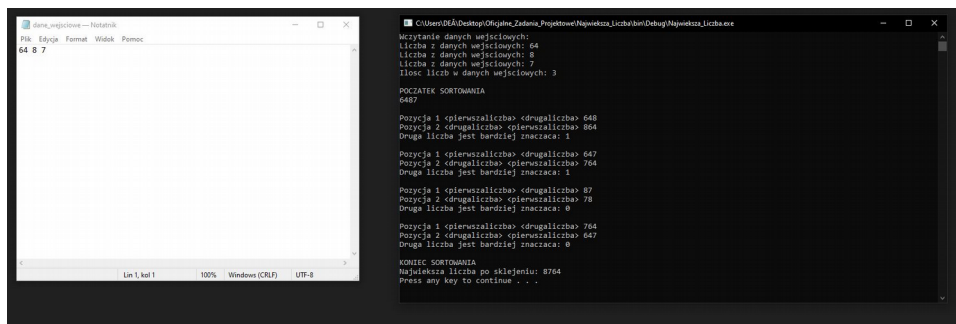
pozycja 1 <pierwszaliczba> <drugialiczba> 8110
pozycja 2 <drugialiczba> <pierwszaliczba> 1081
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 981
pozycja 2 <drugialiczba> <pierwszaliczba> 819
Druga liczba jest bardziej znacząca: 0

pozycja 1 <pierwszaliczba> <drugialiczba> 8110
pozycja 2 <drugialiczba> <pierwszaliczba> 1081
Druga liczba jest bardziej znacząca: 0

KONIEC SORTOWANIA
Najwieksza liczba po sklejeniu: 98110
Press any key to continue . . .
```


Test 4:



```
64 8 7

Wczytanie danych wejściowych:
Liczba 1 danych wejściowych: 64
Liczba 2 danych wejściowych: 8
Liczba 3 danych wejściowych: 7
Liczba 4 danych wejściowych: 3
Liczba 5 danych wejściowych: 3
POCZĄTEK SORTOWANIA
6487
Porównanie 1 (pierwsza liczba) < (druga liczba) 648
Porównanie 2 (druga liczba) < (trzecia liczba) 876
Druga liczba jest bardziej znacząca: 1
Porównanie 1 (pierwsza liczba) < (czwarta liczba) 647
Porównanie 2 (druga liczba) < (piąta liczba) 764
Druga liczba jest bardziej znacząca: 1
Porównanie 1 (pierwsza liczba) < (szesta liczba) 87
Porównanie 2 (druga liczba) < (siódma liczba) 78
Druga liczba jest bardziej znacząca: 0
Porównanie 1 (pierwsza liczba) < (osmia liczba) 764
Porównanie 2 (druga liczba) < (dziewięta liczba) 647
Druga liczba jest bardziej znacząca: 0
KONIEC SORTOWANIA
Największa liczba po skiejeniu: 8764
Press any key to continue . . .
```

7. Podsumowanie

Algorytm ma za zadanie skłajać całość z podanych liczb w największą możliwą. Program ten wykorzystuje do tego dzielenie zwykłe przez dziesięć aby porównać cyfry pierwsze oraz sortowanie bąbelkowego do porównania ze sobą kolejnych liczb w tablicy.